

LISTING OF CLAIMS

The following is a copy of Applicant's claims that identifies language being added with underlining ("____") and language being deleted with strikethrough ("——"), as is applicable:

1. (Previously Presented) A remote object invocation method for invoking a method of a remote object of a remote server; the method comprising the steps of:

- producing remote object data associated with the remote object to discover an object interface dynamically;
- interpretatively establishing a proxy object of a local client computer using the remote object data at runtime of client software, the proxy object bearing an associated proxy method corresponding to the method of the remote object;
- invoking, in response to an action of the client software, the proxy object method;
- conveying invocation data associated with the invocation of the proxy method to the remote object;
- invoking, in response to the invocation data, the method of the remote object;
- and
- returning invocation result data to the client software via the proxy object.

2. (Original) A remote objection invocation method as claimed in claim 1 in which the step of producing the remote object data comprises the step of introspecting the remote object to produce introspection data and in which the remote object method data comprises the introspection data produced by said introspecting.

3. (Previously Presented) A method as claimed in claim 2 further comprising the step of creating an object descriptor for the remote object.

4. (Previously Presented) A method as claimed in claim 3, in which the step of creating the object descriptor for the remote object comprises the step of storing the object descriptor in a cache for later retrieval, the later retrieval being responsive to a request to create an instance of the remote object.

5. (Previously Presented) A method as claimed in claim 4, further comprising the step of determining whether or not an instance of the remote object has been created and in which the step of creating the object descriptor is responsive to a determination that an instance of the remote object has not been created.

6. (Previously Presented) A method as claimed in claim 5, further comprising the step of generating location data to facilitate location of a corresponding instance of the remote object.

7. (Original) A method as claimed in claim 6 in which the step of generating the location data comprises the step of generating an identifier associated with the corresponding instance of the remote object.

8. (Previously Presented) A method as claimed in claim 7, further comprising the step of instantiating the remote object.

9. (Original) A method as claimed in claim 8 in which the step of instantiating the remote object is performed in response to a determination that the remote object has a predetermined characteristic.

10. (Original) A method as claimed in claim 9 in which the predetermined characteristic is that the remote object is stateless.

11. (Original) A method as claimed in claim 9 in which the predetermined characteristic is that the remote object is stateful.

12. (Previously Presented) A method as claimed in claim 11 in which the step of instantiating is performed in response to initialisation of a system hosting the remote object.

13. (Previously Presented) A method as claimed in claim 12, further comprising, prior to the step of instantiating, the step of determining whether or not remote object is permitted to be instantiated and in which the step of instantiating is performed in response to determining that the remote object is permitted to be instantiated.

14. (Previously Presented) A method as claimed in claim 13, further comprising the step of encapsulating the remote object data within a data structure.

15. (Original) A method as claimed in claim 14 in which the data structure is an XML file.

16. (Previously Presented) A remote object invocation method for invoking a method of a remote object of a remote server; the method comprising the steps of introspecting the remote object to produce introspection data associated with the method, the introspection data including an interface description for the remote object; interpretatively processing the introspection data to establish a proxy object of a local client computer bearing an associated proxy method for the remote object method; invoking, in response to an action of client software, the proxy object method; conveying invocation data associated with the proxy object method to the remote object; invoking, in response to the invocation data, the method of the remote object; and returning invocation result data to the client software via the object proxy.

17. (Previously Presented) A method of invoking, from a first computer, a remote object located on a second computer, the method comprising the steps of introspecting, at the second computer, the remote object to identify at least one of a method, property and event thereof to produce introspection data that describes any such identified methods, properties and events; transmitting the introspection data, via a first transport mechanism, to the first computer; creating, at the first computer, a proxy object from the introspection data; invoking, at the first computer, a method of the proxy object; transmitting, from the first computer to the second computer, remote method invocation data, via a second transport mechanism, the remote method invocation data comprising at least an indication of the remote object and the method, property and event thereof to be invoked; receiving, at the second computer, via the second transport mechanism, the remote method invocation data, extracting, at the second computer, the remote method invocation data from the second transport data structure; invoking, at the second computer, the method, property or event of the remote object identified by the remote object invocation data; transmitting, from the second computer to the first computer, via a third transport mechanism, a return object or data representing the results of the invocation of the method, property or event of the remote object; and extracting, at the first computer, the return object or data from the third transport mechanism; wherein the step of creating, at the first computer, the proxy object from the introspection data comprises the step of interpretatively parsing the introspection data and interpretatively constructing the proxy object.

18. (Original) A method as claimed in claim 17 in which the transport mechanisms comprises a combination of an XML file bearing the information to be conveyed and the HTTP.

19. (Previously Presented) A data processing method to invoke a method of a remote object of a remote server; the method comprising the steps of producing remote object data associated with the remote object to discover an object interface dynamically; and interpretatively establishing a proxy object at a local client computer using the remote object data at runtime of client software invoking an associated proxy object method, the proxy object bearing the associated proxy method corresponding to the method of the remote object.

20. (Original) A data processing method as claimed in claim 19 further comprising the steps of invoking, in response to an action of client software, the proxy object method; and conveying invocation data associated with the invocation of the proxy method to the remote object.

21. (Original) A data processing method as claimed in claim 20, further comprising the steps of invoking, in response to the invocation data, the method of the remote object; and returning invocation result data to the client software via the proxy object.

22. (Currently Amended) A system comprising means a computer configured to implement a method as claimed in claim 1.

23. (Previously Presented) A remote object invocation system to invoke a method of a remote object of a remote server; the system comprising a local computer, wherein the local computer comprises an inspector to produce remote object data associated with the remote object, the remote object data including an interface description for the remote object; an interpreter to interpretatively establishing a proxy object using the remote object data at runtime of client software of the local computer; the proxy object bearing an associated proxy method corresponding to the method of remote object; a local invocation means to invoke, in response to an action of the client software, the proxy object method; and a carrier to convey invocation data associated with the invocation of the proxy method to the remote object.

24. (Original) A remote object invocation system as claimed in claim 23 further comprising a remote invocation means to invoke, in response to the invocation data, the method of the remote object; and means to return invocation result data to the client software via the proxy object.

25. (Previously Presented) A remote object server hosting a remote object; the server comprising means to generate introspection data associated with the remote object, the introspection data including an interface description for the remote object, and means to output the introspection data for use by a client in interpretatively creating a proxy object using the introspection data.

26. (Previously Presented) A client computer hosting a receiver for receiving introspection data associated with a remote object of a remote server, the introspection data including an interface description for the remote object, and an interpreter for interpretatively creating a proxy object, using the introspection data, for invocation by an application executable at the client computer.

27. (Currently Amended) A computer readable medium encoded with a computer program for implementing a method as claimed in claim 1, wherein the computer readable medium is a physical structure and the computer program is executed by a computer and causes the computer to perform the method.

28. (Original) A computer program product comprising computer readable storage storing a computer program as claimed in claim 27.